STRUCT

- For Mainframes
- For UNIX/OpenVMS/Windows

STRUCT for Mainframes

```
STRUCT  [(n)]
```

The STRUCT command serves two purposes:

- You can use it to perform structural indentation of a source program.
- Various display features make the structure of a program clear to you, thus allowing you to detect any structural inconsistencies.

The following types of statements are affected by the STRUCT command:

- processing loops (READ, FIND, FOR, etc.),
- conditional statement blocks (AT BREAK, IF, DECIDE FOR, etc.),
- DO/DOEND statement blocks,
- DEFINE DATA blocks,
- inline subroutines.

When you enter the STRUCT command, the STRUCT menu will be displayed. It offers the following functions:

- Generate Structured Source into Work Area
- Display Structure of Source
- Print Structure of Source
- Write Structure of Source into Work Area

Generate Structured Source into Work Area

With this function, you can have a source program indented so that the indentation of source-code lines reflects the structure of the program.

This function is the same as that of the editor command STRUCT.

Indentation will take the source-code line length into consideration; that is, a line to be indented will not be shifted beyond the right margin; if "correct" indentation would require a line to be shifted beyond the right margin, it will only be moved as far to the right as possible, but not beyond the margin.

With the Generate function, you can specify the following options:

Field	Explanation
Source Name	In this field, you enter the name of the source you wish to be structurally indented. The specified source will then be read from the system file into the work area and indented. If you specify no source name, the object currently in the work area of the editor will be indented. If the work area is empty, you must specify a source name.
Shift setting	In this field, you can enter the number of positions (from 1 to 9) by which source-code lines are to be indented. By default, indentation is by 2 positions.
Align Comments	Y Each comment line will be indented as far as the statement line above it; except comment lines which begin at the beginning of a line, these will be not be indented. N Comment lines will not be indented. L Comment lines will be aligned left-justified.
Display Messages	Y A message indicating that the structured program has been generated into the work area and a list of any source-code lines that could not be "correctly" indented (see above) will be displayed. N No such messages will be displayed.
Return to STRUCT	Y You will be returned to the STRUCT menu after the Generate function has been executed. N You will be returned to the screen from where you issued the STRUCT command after the Generate function has been executed.

Note:
Indentation is performed differently for a reporting-mode program than for a structured-mode program.

Partial Indentation

You can exclude sections of your program source from structural indentation by using the special statements "/*STRUCT OFF" and "/*STRUCT ON". These must be entered at the beginning of a source-code line. The source-code lines between these two statements will remain as they are when you execute the Generate function.

Example of Structural Indentation:

Program before being structurally indented:

```
0010 DEFINE DATA LOCAL
0020 1 EMPL VIEW OF EMPLOYEES
0030 2 PERSONNEL-ID
0040 2 FULL-NAME
0050 3 FIRST-NAME
0060 3 NAME
0070 1 VEHI VIEW OF VEHICLES
0080 2 PERSONNEL-ID
0090 2 MAKE
0100 END-DEFINE
0110 FIND EMPL WITH NAME = 'ADKINSON'
0120 IF NO RECORDS FOUND
0130 WRITE 'NO RECORD FOUND'
0140 END-NOREC
0150 FIND (1) VEHI WITH PERSONNEL-ID = EMPL.PERSONNEL-ID
0160 DISPLAY EMPL.PERSONNEL-ID FULL-NAME MAKE
0170 END-FIND
0180 END-FIND
0190 END
```

The same program after the function Generate Structured Source has been applied to it:

```
0010 DEFINE DATA LOCAL
0020 1 EMPL VIEW OF EMPLOYEES
0030   2 PERSONNEL-ID
0040   2 FULL-NAME
0050     3 FIRST-NAME
0060     3 NAME
0070 1 VEHI VIEW OF VEHICLES
0080   2 PERSONNEL-ID
0090   2 MAKE
0100 END-DEFINE
0110 FIND EMPL WITH NAME = 'ADKINSON'
0120   IF NO RECORDS FOUND
0130     WRITE 'NO RECORD FOUND'
0140   END-NOREC
0150   FIND (1) VEHI WITH PERSONNEL-ID = EMPL.PERSONNEL-ID
0160     DISPLAY EMPL.PERSONNEL-ID FULL-NAME MAKE
0170   END-FIND
0180 END-FIND
0190 END
```

Display Structure of Source

With this function, you can display the source code of an object along with several items of information which make the structure of the object clear.

With the Display function, you have the following options:

Field	Explanation
Source Name	In this field, you enter the name of the source you wish to be displayed. The specified source will then be read from the system file and displayed. If you specify no source name, the object currently in the work area of the editor will be displayed. If the work area is empty, you must specify a source name.
Display Compressed	Y Source-code lines on the same structural level will not be displayed. Only those lines will be displayed which cause a change in the structure table on the right-hand side of the screen. From the gap in the sequence of line numbers you can tell how many lines are not shown between two given lines displayed. N All source-code lines will be displayed.
Return to STRUCT	Y You will be returned to the STRUCT menu after the Display function has been executed. N You will be returned to the screen from where you issued the STRUCT command after the Display function has been executed.

The following information is displayed:

Line Numbers	For every statement which closes a statement block, the source-code line number of the corresponding statement which initiates the statement block will be displayed to the left of the source code.
Structure Table	To the right of the source code, a table is displayed, which contains indicators for open statement blocks. For each open statement block, a single letter is displayed. The different letters refer to different types of statements (for an explanation of the letters, press PF1). Any structural inconsistency in the source code is indicated by a message being displayed in the structure table.

Example of Display with Structure Information:

```
11:49:23 - Structured Source ABC in Library XYZ -                    2001-01-12
  0010      DEFINE DATA LOCAL                                    *O
  0020      1 EMPL VIEW OF EMPLOYEES                             *O
  0030        2 PERSONNEL-ID                                     *O
  0040        2 FULL-NAME                                        *O
  0050          3 FIRST-NAME                                     *O
  0060          3 NAME                                           *O
  0070      1 VEHI VIEW OF VEHICLES                              *O
  0080        2 PERSONNEL-ID                                     *O
  0090        2 MAKE                                             *O
  0100 0010 END-DEFINE                                           *O
  0110      FIND EMPL WITH NAME = 'ADKINSON'                     *F
  0120         IF NO RECORDS FOUND                               *FJ
  0130           WRITE 'NO RECORD FOUND'                         *FJ
  0140 0120   END-NOREC                                          *FJ
  0150         FIND (1) VEHI WITH PERSONNEL-ID = EMPL.PERSONNEL-I *FF
  0160           DISPLAY EMPL.PERSONNEL-ID FULL-NAME MAKE        *FF
  0170 0150   END-FIND                                           *FF
  0180 0110 END-FIND                                             *F
  0190      END                                                  *
  PF1=Help, PF2=Menu, PF3=Exit, PF6=Top, PF12=Cancel.
```

The current content of the work area is not affected by the displayed source.

Print Structure of Source

With this function, you can print the source code of an object along with its structural information.

The Print function corresponds to the function Display Structure of Source, only the output is not displayed on the screen but sent to a printer.

With the Print function, you have the same options as with the Display function.

Write Structure of Source into Work Area

With this function, you can read a source from the system file and write it into the editor work area together with its structure information, plus several lines (line numbers 0000) at the beginning of the source, which explain the structure information.

With the Write function, you have the same options as with the function Display Structure of Source, except that you **must** specify a Source Name.

The source and its structure information are written as text into the work area, and can be edited with the system command EDIT.

STRUCT for UNIX/OpenVMS

```
STRUCT [(n)]
```

The STRUCT command performs structural indentation of a source program.

The following types of statements are affected by the STRUCT command:

- processing loops (READ, FIND, FOR, etc.),
- conditional statement blocks (AT BREAK, IF, DECIDE FOR, etc.),
- DO/DOEND statement blocks,
- DEFINE DATA blocks,
- inline subroutines.

With this function, you can have a source program indented so that the indentation of source-code lines reflects the structure of the program.

Note:
Indentation is performed differently for a reporting-mode program than for a structured-mode program.

Partial Indentation

You can exclude sections of your program source from structural indentation by using the special statements "/*STRUCT OFF" and "/*STRUCT ON". These must be entered at the beginning of a source-code line. The source-code lines between these two statements will remain as they are when you execute the Generate function.

Example of Structural Indentation:

Program before being structurally indented:

```
0010 DEFINE DATA LOCAL
0020 1 EMPL VIEW OF EMPLOYEES
0030 2 PERSONNEL-ID
0040 2 FULL-NAME
0050 3 FIRST-NAME
0060 3 NAME
0070 1 VEHI VIEW OF VEHICLES
0080 2 PERSONNEL-ID
0090 2 MAKE
0100 END-DEFINE
0110 FIND EMPL WITH NAME = 'ADKINSON'
0120 IF NO RECORDS FOUND
0130 WRITE 'NO RECORD FOUND'
0140 END-NOREC
0150 FIND (1) VEHI WITH PERSONNEL-ID = EMPL.PERSONNEL-ID
0160 DISPLAY EMPL.PERSONNEL-ID FULL-NAME MAKE
0170 END-FIND
0180 END-FIND
0190 END
```

The same program after the function Generate Structured Source has been applied to it:

```
0010 DEFINE DATA LOCAL
0020 1 EMPL VIEW OF EMPLOYEES
0030   2 PERSONNEL-ID
0040   2 FULL-NAME
0050     3 FIRST-NAME
0060     3 NAME
0070 1 VEHI VIEW OF VEHICLES
0080   2 PERSONNEL-ID
0090   2 MAKE
0100 END-DEFINE
0110 FIND EMPL WITH NAME = 'ADKINSON'
0120   IF NO RECORDS FOUND
0130     WRITE 'NO RECORD FOUND'
0140   END-NOREC
0150   FIND (1) VEHI WITH PERSONNEL-ID = EMPL.PERSONNEL-ID
0160     DISPLAY EMPL.PERSONNEL-ID FULL-NAME MAKE
0170   END-FIND
0180 END-FIND
0190 END
```

The parameter (n) may be supplied to specify the number of spaces used for identation. If (n) is unspecified, identation is set to 2.

Example:

```
STRUCT (5)
```